

Relatório Técnico — Sistema de Leilão TCP (Cliente-Servidor)

1. Especificação do Protocolo

a. Objetivo da Aplicação

O sistema simula um leilão de itens entre múltiplos clientes conectados a um servidor central. O objetivo é permitir que usuários possam se autenticar, entrar em uma sala de leilão, fazer lances e acompanhar o resultado em tempo real. O sistema segue a arquitetura cliente-servidor, usando comunicação TCP persistente.

b. Características do Protocolo

- Arquitetura: Cliente-Servidor
- Protocolo: TCP (sockets)
- Comunicação: bidirecional, orientada a texto
- Persistência: conexões mantidas ativas até logout ou encerramento
- Leilão: inicia automaticamente com 2 usuários, termina após tempo fixo

c. Lista de mensagens (Cliente → Servidor)

Comando	Descrição
LOGIN	Autentica o cliente
INFO	Retorna o nome do usuário autenticado
LOGOUT	Encerra a sessão do usuário
ENTRAR_LEILAO	Entra na fila de participantes
LANCE	Envia um lance para o leilão
PING	Teste de conexão

d. Lista de mensagens (Servidor → Cliente)

Resposta	Descrição
LOGIN_OK / LOGIN_FAIL	Resultado da autenticação
Você já está logado.	Evita múltiplos logins
Você entrou no leilão.	Confirma entrada
LEILAO_INICIO	Início automático do leilão
NOVO_LANCE	Broadcast de novo lance
LANCE_REJEITADO	Lance inválido ou abaixo do atual
LEILAO_FIM	Resultado final
Você não está logado.	Acesso não autorizado
PONG	Resposta ao PING

e. Formato das mensagens

Mensagens são de texto puro, terminadas por `\n`. São simples comandos com argumentos separados por espaço. Exemplo:

...

```
LANCE 1300
LEILAO_INICIO Notebook_Dell 1000.00 30
```

...

f. Valores e Regras

- Usuários válidos: definidos no código (``joao 123``, ``maria abc``, ...)
- Leilão inicia com 2 participantes
- Tempo fixo por rodada (30s)
- Lance deve ser maior que o anterior
- Lances ignorados após ``LEILAO_FIM``
- Sessão por socket (um login por cliente)

g. Observações Técnicas

- Implementado em linguagem C
- Uso de `select()` no servidor para multiplexação
- Modularização em `server.c`, `client.c`, `auction.c`, `users.c`
- Includes e headers organizados
- `makefile` para build automático
- Testado com múltiplos clientes simultâneos
- Tolerância a falhas de cliente (Ctrl+C não derruba o servidor)
- Pronto para rodadas consecutivas de leilão